

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: MEDIA GATEWAY

APPLICANT: NICK RAMIREZ

CERTIFICATE OF MAILING BY EXPRESS MAIL

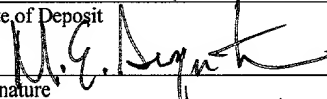
Express Mail Label No. EL258933616US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

Date of Deposit

10-15-01

Signature



Typed or Printed Name of Person Signing Certificate

Mike Augustine

MEDIA GATWAY

TECHNICAL FIELD

5 This invention relates generally to a media gateway and, more particularly, to a media gateway that includes a digital signal processor that packetizes data and a network processor that directs the resulting data packets to a network.

BACKGROUND

10 To implement voice-over Internet Protocol (VOIP), audio information is packetized and the resulting data packets are transmitted over a network, such as the Internet, between two devices. Streaming video is implemented in much the same way.
15 That is, video information is packetized and the resulting data packets are transmitted over a network between two devices. A network processor may be used to regulate the flow of packet traffic, including VOIP or streaming video packets, on the network. For example, a network processor may be used
20 to redirect packet traffic in order to reduce congestion on the network or to re-prioritize the data packets.

DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a media gateway which interfaces one or more devices to a network.

Fig. 2 is a flowchart showing processing of data packets that occurs in the media gateway.

DESCRIPTION

Referring to Fig. 1, a computer network 10 is shown. Computer network 10 includes the following hardware: media gateway 12, media gateway controller 14 (which includes network interface cards (NICs) 16 and 18 and control module 15), T1 local switch 20, and signaling gateway 22.

In this embodiment, data is transferred from a T1 line 24, through T1 local switch 20, to media gateway 12. The data may include voice information to implement VOIP and/or video information to implement streaming video. Other types of data may be transmitted along with, or instead of, this data.

In the case of VOIP, a user activates (e.g., picks-up) a handset 26, which is connected to T1 line 24. Voice and signaling information is transmitted from handset 26 to T1 local switch 20. The voice information comprises the content of the VOIP. The signaling information identifies the location of the handset and the target of the call. In more

detail, the signaling information includes the network (e.g., IP) address of handset 26 and the telephone number of the person being called. In this context, "telephone number" can mean a standard nine-digit (with area code) telephone number or any other type of identifier for use with VOIP.

From T1 local switch 20, the voice information is transmitted to media gateway 12 and the signaling information is transmitted to signaling gateway 22. Signaling gateway 22 is an interface between the network 28 carrying the signaling information and a local bus 30. NIC 16 is an interface between local bus 30 and media gateway controller 14.

Media gateway controller 14, in particular control module 15, receives the signaling information (e.g., a number) via signaling gateway 22 and NIC 18 and translates the signaling information to network addressing information. In this embodiment, the network addressing information is a device IP address; however, the media gateway described herein is not limited to use with IP addresses.

Media gateway controller 14 provides the network addressing information to media gateway 12 via a peripheral component interconnect (PCI) bus 32. Media gateway 12 sends VOIP packets to device(s) identified by the network addressing information received from media gateway controller 14, as

described in more detail below. Media gateway controller 14 also uses the signaling information to contact a device on a remote network 34, such as the Internet, and open a channel to that device. NIC 18 interfaces between media gateway controller 14 and remote network 34.

To open a channel to a device, media gateway controller 14 may send data packets to the IP address of a remote device. The data packets instruct the device to ready itself to receive VOIP data. The device may then issue a return confirmation that a channel has been opened and that it is ready to receive VOIP data packets. The VOIP data packets are sent to the remote device from media gateway 12, as described below.

In this embodiment, media gateway 12 includes T1 media access control (MAC) interface 36, network processor MAC interface 38, network processor 40, high-speed memory 42, and digital signal processor (DSP) 44. Media gateway 12 receives pulse-code-modulated (PCM) data from T1 local switch 20 and network addressing information (e.g., device IP address(es)) from media gateway controller 14. Media gateway 12 generates data packets from the data (in this case, VOIP data packets, but it may also generate streaming video, or other types, of

data packets). Media gateway 12 sends the data packets to device(s) specified by the network addressing information.

Referring to Fig. 2, T1 MAC interface 36 receives (201) the PCM data from T1 local switch 20. T1 MAC interface 36
5 transmits the PCM data to network processor 40. Network processor 40 executes one or more packet engines (not shown), which may be implemented in computer code, to store and route data packets. In this case, the packet engines receive (202)
10 the PCM data from T1 MAC interface 36 and pass the PCM data to DSP 44. The packet engines may queue the data in high-speed memory 42 prior to passing the data to DSP 44.

DSP 44 receives (203) the PCM data from the packet engines, compresses (204) the data, and generates (205) data packets that correspond to the compressed data. For example,
15 the PCM data may be in the form of T1 data packets. In this example, DSP 44 compresses the T1 data packets using a compression process, such as G.711 or G.728, and generates a different type of data packet, such as an asynchronous transfer mode (ATM) data packet, that contains the compressed
20 data. DSP 44 may generate any type of data packet that is capable of being transmitted over a network. Any type of compression process may be used, not just those listed above.

DSP 44 may also encrypt the data packets before the data packets are sent back to the packet engine(s) in network processor 40. Any type of encryption process may be used, such as PGP (Pretty Good Privacy) encryption. It is noted
5 that encryption is optional and not necessary.

DSP 44 sends (206) the compressed (and maybe encrypted) data packets back to network processor 40. The packet engines receive the data packets and may queue the data packets in high-speed memory 42, if necessary, e.g., to reduce congestion on external network 34 and/or prioritize the data packets for
10 transmission to external network 34. Network processor 40 also receives network addressing information (e.g., IP addresses) from media gateway controller 14. The network addressing information identifies one or more devices to which
15 data packets are to be sent (i.e., devices with which media gateway controller 14 opened a channel - see above).

The packet engine(s) in network processor 40 direct (207) the compressed (and maybe encrypted) data packets to the device(s) on external network 34 that are identified by the
20 addressing information. That is, the packet engines send the data packets to those devices via network processor MAC interface 38. Network processor MAC interface 38 may contain

a controller or the like to regulate the flow of data packets between the packet engines and external network 34.

A remote device (not shown) on external network 34 that receives the data packets may respond in kind. The data packets from the remote device will travel through network processor MAC interface 38, to network processor 40, and then to DSP 44. DSP 44 decompresses and decrypts the data packets, if necessary, and converts the data packets to PCM data in a T1-format data packet. DSP 44 sends the data packets to network processor 40, which transmits them through T1 MAC interface 36 and T1 local switch 20 to handset 26.

A remote device may also set up a channel to another device via media gateway 12. In this case, the remote device sends signaling information to media gateway 12 via NIC 18 and media gateway controller 14. Media gateway 12 uses the signaling information to open up a channel between the other device and the remote device and, thereafter, to send data packets received from the remote device to the other device.

Media gateway 12 is described above in the context of VOIP. However, it has other uses, as noted. For example, media gateway 12 may be used to transmit streaming video. In this case, the signaling information includes a network address of a device to transmit the streaming video and a

network address of a device to receive the streaming video (instead of a telephone number). Media gateway controller 14 sets up a channel between the two devices and the resulting video data packets are processed and transmitted in the same manner as the VOIP data packets described above.

Media gateway 12 also may be used to set up a secure virtual private network (VPN). In this example, the information sent between a sending device (in place of handset 26) and a remote receiving device comprises network addresses and encryption protocols necessary to set up the VPN. Otherwise, the data packet processing and transfer is the same as described in the example above. T1 MAC 36 can be replaced with an Internet MAC to implement the VPN. The Internet MAC and media gateway 10 can also be used to support a firewall.

The packet engines in network processor 40 may be implemented in computer programs executing on programmable computers or other machines that each includes a network processor and a storage medium readable by the processor.

Each such program may be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. However, the programs can be implemented in assembly or machine language. The language may be a compiled or an interpreted language.

Each computer program may be stored on an article of manufacture, such as a CD-ROM, hard disk, or magnetic diskette, that is readable by network processor 40 to process data packets in the manner described above. The packet engines may also be implemented as a machine-readable storage medium, configured with one or more computer programs, where, upon execution, instructions in the computer program(s) cause the network processor to operate as described above.

Likewise DSP 44 contains computer code to implement its functions described above. The computer code may be implemented in computer programs executing on programmable computers or other machines that each include a DSP and a storage medium readable by the DSP.

Each such program may be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. However, the programs can be implemented in assembly or machine language. The language may be a compiled or an interpreted language.

Each computer program may be stored on an article of manufacture, such as a CD-ROM, hard disk, or magnetic diskette, that is readable by DSP 44 to process data packets in the manner described above. The computer code may also be implemented as a machine-readable storage medium, configured

with one or more computer programs, where, upon execution, instructions in the computer program(s) cause DSP 44 to operate in the manner described above.

The invention is not limited to the embodiments described above. For example, the invention is not limited to the specific hardware and software configurations described above. The invention is not limited to the protocols (e.g., encryption) described above. A standard, appropriately programmed, microprocessor or other controller could be used in lieu of network processor 40 and/or DSP 44. It is further noted that DSP 44 can reside on network processor 40, even though the two are shown separate in Fig. 1.

Other embodiments not described herein are also within the scope of the following claims.

What is claimed is: